Efficient Geometric Disassembly of Multiple Components from an Assembly Using Wave Propagation

Hari Srinivasan
Doctoral Candidate,
e-mail: hari@smartcad.me.wisc.edu

Rajit Gadh
Professor,
e-mail: gadh@engr.wisc.edu

I-CARVE Lab,
Department of Mechanical Engineering,
University of Wisconsin-Madison,
1513 University Avenue, #347,
Madison, WI 53706
(http://icarve.me.wisc.edu)

This paper analyzes the problem of disassembling multiple selected components from an assembly, defined as selective disassembly, and presents algorithms for efficient disassembly analysis of geometric models. Applications for selective disassembly include assembling, maintenance and recycling. A new approach called 'Disassembly Wave Propagation' is proposed to determine a selective disassembly sequence with minimal component removals from an assembly. This approach defines: (i) disassembly waves to topologically arrange the components denoting the disassembly order and (ii) intersection events between the waves to determine the selective disassembly sequences. In order to evaluate a minimal removal sequence in a feasible computation time, algorithms are proposed that prioritize and process the intersection events based on the order in which they occurred. The proposed algorithms analyze selective disassembly from the geometric perspective and are applicable for both two-dimensional and three-dimensional product assemblies. [S1050-0472(00)01402-1]

Keywords: Design for Disassembly, Selective Disassembly, Disassembly Sequence, Wave Propagation, Assembly Planning and Product Design

1 Introduction

A product designed for disassembly can be taken apart easily to support applications such as assembling, maintenance and recycling. For these applications, the disassembly analysis involves evaluating a disassembly sequence (**S**) from a geometric model of an assembly (**A**). In general, two categories of problems exist in disassembly sequence analysis:

- *Complete Disassembly* (**CD**) involves disassembling all the components in **A** to obtain **S**. For example, to disassemble all the components in Fig. 1, $\mathbf{S}=\{\mathbf{C_9},\mathbf{C_7},\mathbf{C_1},\mathbf{C_2},\mathbf{C_8},\mathbf{C_6},\mathbf{C_4},\mathbf{C_3},\mathbf{C_5}\}$.
- *Selective Disassembly* (**SD**) involves disassembling a subset of components (**C**) from **A** to obtain **S**. For example, in Fig. 1, to disassemble $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$, $\mathbf{S}=\{\mathbf{C_1},\mathbf{C_4},\mathbf{C_3},\mathbf{C_5}\}$.

An application for **CD** is assembling, since reversing **S** potentially can yield an assembly sequence [1]. For example, in Fig. 1, the reverse of **S** yields one potential assembly sequence $\{\mathbf{C_5},\mathbf{C_3},\mathbf{C_4},\mathbf{C_6},\mathbf{C_8},\mathbf{C_2},\mathbf{C_1},\mathbf{C_7},\mathbf{C_9}\}$.

SD is an important research area for applications like maintenance, assembling and recycling [2]. These applications usually require assembly/disassembly of a subset of components from **A** rather than the entire assembly, which provides a motivation for **SD**. For example, automotive engine maintenance requires the **SD** of the engine and not the **CD** of the entire vehicle. Another use of **SD** is in recycling applications that require the removal of some high-valued components, such as **SD** of an Instrument Panel from a car assembly. Therefore, evaluating the design for **SD** is an important area of research in product development [3].

1.1 Multiple Component Disassembly. This paper analyzes the following multiple-component **SD** problem: Given **A** of **n** components and **C** of **s** components, automatically determine **S** to disassemble **C** with minimal component removals. The objective of minimal component removals is appropriate, since for 1-disassemblable components (a 1-disassemblable component requires a single linear motion to be removed from **A**), the objective becomes minimizing the disassembly motions (operations), which is a measure of difficulty of disassembling [1]. Moreover, the product design for manufacture suggests simple motions and easier separation of components [4,5] for product assembling/maintenance/recycling. Therefore, **S** with minimal component removals is defined as an Optimal Sequence (**OS**).

To illustrate the multiple-component **SD** problem, consider **A** in Fig. 1 with the requirement to disassemble $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$. Let $\mathbf{n}_S$ denote the number of components in **S**. For $\mathbf{C}=\{\mathbf{C_3}\}$, an $\mathbf{OS}=\{\mathbf{C_2},\mathbf{C_3}\}$ with $\mathbf{n_S}=2$. For $\mathbf{C}=\{\mathbf{C_5}\}$, two **OS**'s with $\mathbf{n_S}=3$ exist: $\{\mathbf{C_7},\mathbf{C_6},\mathbf{C_5}\}$ and $\{\mathbf{C_1},\mathbf{C_4},\mathbf{C_5}\}$. Aggregating these two sequences (one with $\mathbf{C}=\{\mathbf{C_3}\}$ and another with $\mathbf{C}=\{\mathbf{C_5}\}$) for $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$ results in $\mathbf{S}=\{\mathbf{C_2},\mathbf{C_3},\mathbf{C_7},\mathbf{C_6},\mathbf{C_5}\}$ and $\{\mathbf{C_2},\mathbf{C_3},\mathbf{C_1},\mathbf{C_4},\mathbf{C_5}\}$ with $\mathbf{n_S}=5$. However, for $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$ a better solution exists: $\mathbf{OS}=\{\mathbf{C_1},\mathbf{C_4},\mathbf{C_3},\mathbf{C_5}\}$ with $\mathbf{n_S}=4$. This example illustrates that aggregating individual **OS** for $\mathbf{C_x}\in\mathbf{C}$ results in **S** for **C**, which in general is not an **OS**. Therefore a separate approach for multiple component **SD** is required. The current research presents an efficient method to determine an **OS** for **SD** analysis. However, prior to presenting the current research, some related work in disassembly analysis and potential approaches for **SD** are presented.

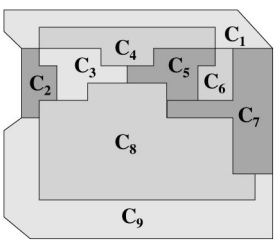

Fig. 1 Test assembly to illustrate SD problem

Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received Sept. 1999; revised Apr. 2000. Associate Technical Editor: M. A. Ganter.

1.2 Prior Approaches. One potential approach to determining **OS** for **SD** is to enumerate (exhaustively) all the possible sequences and to select a sequence with minimal removals. However, this analysis is computationally expensive (exponential to the number of components in **A**) and is not recommended.

Another possible approach is to determine **OS** for **C** from a **CD** sequence. Several representations/approaches [6] allow evaluation of **CD** sequences: (i) *Assembly Sequence Diagram* [7], which represents the ability or inability to assemble a part to a subassembly; (ii) *AND/OR Graph* [1], which establishes conditions and precedence relationships between components; (iii) *Abstract Liaison Graph* [8], which represents the stability of part interconnections and the directional constraints of the motions that bring two parts together; (iv) *Non-Directional Blocking Graph* [9], which describes part interactions from the blocking nature of components by utilizing the concept of graph partitioning; (v) *Geometrical Constraints* [10,11], which evaluates the ease of disassemblability of components for sequencing. However, the **S** obtained from a **CD**, may not give an optimal solution. For example, **S** can be obtained by recursively disassembling components that are disassemblable in **A** until all the components in **C** are disassembled. To illustrate this, consider **A** in Fig. 1 with $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$. Following the **CD** algorithm, **S** to disassemble **C** is $\{\mathbf{C_9},\mathbf{C_7},\mathbf{C_1},\mathbf{C_2},\mathbf{C_8},\mathbf{C_6},\mathbf{C_4},\mathbf{C_3},\mathbf{C_5}\}$ with $\mathbf{n_S}=9$, but **OS** $=\{\mathbf{C_1},\mathbf{C_4},\mathbf{C_3},\mathbf{C_5}\}$ with $\mathbf{n_S}=4$. Hence, a separate approach for **SD** analysis is required.

Another approach for **SD** is the construction of a Disassembly Tree [12,13], which is designed to model the ''Onion Peeling'' abstraction: recursively disassembling removable components starting from the boundary of **A** and proceeding inwards. The Disassembly Tree approach is proposed for 2.5D objects and the analysis is based on the contact geometry. However, this algorithm is only applicable for assemblies in which every component is disassembled by removing none or one of its mating adjacent. Therefore, this approach is too restrictive for our use.

Most of the previous work on assembly and disassembly planning has focused on **CD**. However, there has been little investigation of **SD** techniques.

1.3 Efficient Selective Disassembly. In the **SD** problem, the requirement is to identify **S** to disassemble **C**. However, apart from the objective that the **SD** analysis should be automatic and analyze 3D geometric models, there are two other important issues: (i) Computationally efficient algorithms and (ii) Optimum **SD** solution. Efficiency and optimality are related and one is usually achieved at the cost of the other. For example, if efficiency is the only issue, then any of the **CD** solutions can be extended for **SD**; however, this results in a non-optimum solution, as discussed above. On the other hand, if optimality is the only issue, then exhaustive enumeration will give an optimum solution; however, this approach is computationally inefficient, as discussed above. Therefore, the current research attempts to provide abstractions that balance the requirements of computational efficiency and optimality, i.e., determining a **SD** solution with fewer component removals that can be computed in a feasible computation time.

2 Disassembly Wave Propagation Approach

Given an assembly of **n** components, **SD** of $1<\mathbf{s}<\mathbf{n}$ components is defined as multiple-component **SD**; where **s** $=$Cardinality (**C**). This section presents a new approach called Disassembly Wave Propagation (**DWP**) for multiple-component **SD**. The motivation for **DWP** is that an optimum solution may be obtained if two or more components are disassembled along a common sequence.

2.1 Assumptions of the Current Research.

1 The relative motions of the components are determined without considering the tools, fixtures or robots required to achieve these motions.

2 Assemblies are assumed to be polyhedral, rigid, frictionless, defined by nominal geometry, and have tightly fitted components.

3 Components are 1-disassemblable and 1-dependent (a component is disassemblable after removing one of its adjacent components). Moreover, locally disassemblable components are assumed to be completely disassembled from **A**.

4 Disassembly sequences are sequential, monotonic, and nondestructive [2].

Assumptions 1–4 are standard assumptions followed by different researchers in automated assembly/disassembly analysis. Assumption 1 requires fixture elements to be modeled based on the sequence determined, or to be modeled as constraints to components [1]. Assumption 2 regarding polyhedral assemblies requires transforming free-form surfaces to planar surfaces. However, the polyhedral assembly assumption is relaxed if the collision detection technique (discussed in Section 4.2) is used to compute the geometric attributes. The 1-disassemblable assumption is utilized because automated disassembly allowing general disassembly motion is computationally expensive [14]. Moreover, the 1-disassemblable assumption is realistic for real world examples [15] and also is reasonable, since design for manufacturing recommends simple motions for disassembly [5].

2.2 Geometric Attributes. The **DWP** approach defines a disassembly wave to topologically arrange $\mathbf{C_i}\in\mathbf{A}$, denoting the disassembly order such that a component in one wave is disassembled after removing its adjacent component in the next wave. This section presents the geometric attributes used in defining the disassembly waves.

• *Disassembly Directions*: Let the $\mathbf{k}^{\text{th}}$ mating face of $\mathbf{C_i}$ and $\mathbf{C_j}$ be represented as $\mathbf{M^k_{i,j}}$. For every mating face $\mathbf{M^k_{i,j}}$, the directions along which $\mathbf{C_i}$ can be locally disassembled relative to $\mathbf{C_j}$ is represented as a set of directions $\mathbf{d^k_{i,j}}$ on a Gaussian Sphere [9] and for 2D it is represented on a Gaussian Circle. For example, Fig. 2 shows the disassembly directions $\mathbf{d^1_{2,1}}$ and $\mathbf{d^2_{2,1}}$ for the mating faces $\mathbf{M^1_{2,1}}$ and $\mathbf{M^2_{2,1}}$ respectively.

• *Accessibility: Accessibility* of $\mathbf{C_i}$ with respect to adjacent component $\mathbf{C_j}$ is defined as the set of directions with which $\mathbf{C_i}$ can move relative to $\mathbf{C_j}$ and is denoted as $\mathbf{AC^j_i}$. For example, Fig. 3 illustrates the accessibility of components. Accessibility of components due to face and thread mating is derived from the nature of components. An example illustrating the computation of accessibility as an intersection of disassembly directions due to face mating is shown in Fig. 2. In Fig. 3, $\mathbf{C_2}$ has a threaded contact with $\mathbf{C_4}$, therefore $\mathbf{AC^2_4}=$NULL, i.e., $\mathbf{C_2}$ must be removed prior to disassembling $\mathbf{C_4}$.

• *Disassemblability*: Disassemblability, $\Delta_\mathbf{i}$, is a binary value that indicates if $\mathbf{C_i}\in\mathbf{A}$ is removable. $\Delta_\mathbf{i}$ is computed as the intersection of all $\mathbf{AC^j_i}$ (where $\mathbf{C_j}$ is the mating adjacent of $\mathbf{C_i}$). For example, in Fig. 3 $\Delta_\mathbf{1}=$TRUE for $\mathbf{C_1}$ and $\Delta_\mathbf{2}=$FALSE for $\mathbf{C_2}$. A disassemblable component is defined as a boundary component and is denoted as $\mathbf{C_b}$.

• *Removal Influence*: Let $\mathbf{MA_i}$ denote the mating adjacents of $\mathbf{C_i}$. For $\mathbf{C_i}\in\mathbf{A}$, let the removal influence of $\mathbf{C_j}\in\mathbf{MA_i}$ on $\mathbf{C_i}$ be

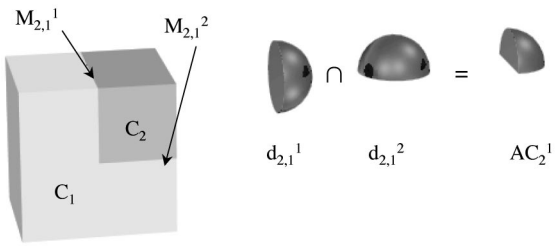

Fig. 2 Mating faces and accessibility directions

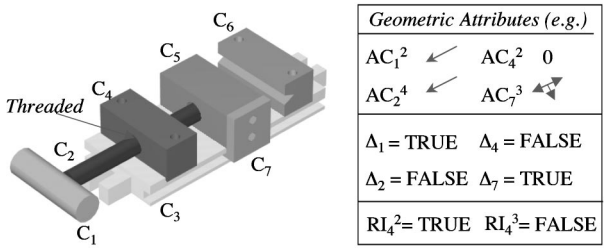

Fig. 3 Machine vice assembly: geometric attributes

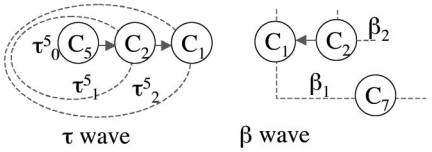

Fig. 5 τ^5 wave and β wave for A in Fig. 3

denoted as $\mathbf{RI}_i^j$. If Δ_i=FALSE and with the removal of $\mathbf{C}_j$ in **A**, Δ_i=TRUE then $\mathbf{RI}_i^j$=TRUE; else $\mathbf{RI}_i^j$=FALSE. For example, in Fig. 3 $\mathbf{RI}_4^2$=TRUE, since Δ_4 is TRUE with the removal of $\mathbf{C}_2$ in **A**. Furthermore, $\mathbf{RI}_4^3$=FALSE, since Δ_4 is FALSE with the removal of $\mathbf{C}_3$ in **A**.

2.3 Disassembly Waves. **SD** of a single component involves determining only the disjoint sequences of **C** [16]. However, multiple-component **SD** analysis involves determining both the disjoint and common sequences of **C**. The **DWP** approach defines two types of disassembly waves to evaluate both the disjoint and common sequences:

- τ waves from **C**, which propagate outwards.
- β waves from the boundary of **A** (the enclosing region of **A** that includes zero components), which propagate inwards.

Let the τ wave of $\mathbf{C}_x \in \mathbf{C}$ be denoted as τ^x and τ_a^x=the set of components in τ^x which are **a** ($>=0$) units away from $\mathbf{C}_x$. Then the propagation of τ^x from τ_{a-1}^x to τ_a^x is defined as follows:

Definition 1 (τ *wave propagation*) For $\mathbf{a}=0$, $\tau_a^x=\{\mathbf{C}_x\}$. For $\mathbf{a}>0$, a τ wave propagation from $\mathbf{C}_i \in \tau_{a-1}^x$ to $\mathbf{C}_j \in \mathbf{MA}_i$ exists if Δ_i=FALSE, $\mathbf{C}_j \notin (\tau_0^x \cup \tau_1^x \ldots \cup \tau_{a-1}^x)$ and $\mathbf{RI}_i^j$=TRUE, then $\mathbf{C}_j \in \tau_a^x$.

Let β_a=the set of components in the β wave which are **a** ($>=0$) units away from the boundary of **A**. Then the propagation of β wave from β_{a-1} to β_a is defined as follows:

Definition 2 (β wave propagation) For $\mathbf{a}=0$, $\beta_a=\{\}$. For $\mathbf{a}=1$, β_a=set of all $\mathbf{C}_b \in \mathbf{A}$. For $\mathbf{a}>1$, β wave propagation from $\mathbf{C}_j \in \beta_{a-1}$ to $\mathbf{C}_i \in \mathbf{MA}_j$ exists if Δ_i=FALSE, $\mathbf{C}_i \notin (\beta_0 \cup \beta_1 \ldots \cup \beta_{a-1})$ and $\mathbf{RI}_i^j$=TRUE, then $\mathbf{C}_i \in \beta_a$.

A disassembly wave is represented by a *Removal influence Graph* (**RG**) whose nodes correspond to components in the disassembly wave and arcs correspond to the removal influence between the components. Figure 4 illustrates τ and β wave propagation. The τ wave from $\mathbf{C}_i$ to $\mathbf{C}_j$, represented as $\mathbf{C}_i \rightarrow \mathbf{C}_j$, implies that $\mathbf{C}_i$ is disassemblable after removing $\mathbf{C}_j$. The β wave from $\mathbf{C}_j \in \beta_{a-1}$ to $\mathbf{C}_i \in \beta_a$ is represented as $\mathbf{C}_i \rightarrow \mathbf{C}_j$, denoting that $\mathbf{C}_i$ is disassemblable after disassembling $\mathbf{C}_j$ (the reason for having reverse logic of the arrow for the β wave in **RG** is to maintain consistency in disassembly ordering).

A τ wave of $\mathbf{C}_x \in \mathbf{C}$ (denoted as τ^x) topologically orders $\mathbf{C}_i \in \mathbf{A}$ with respect to C_x and determines the disassembly ordering for $\mathbf{C}_x$. For example, Fig. 5 illustrates τ wave propagation of $\mathbf{C}_5(\tau^5)$ for **A** in Fig. 3. τ^5 propagates from τ_0^5 to τ_1^5 and then propagates from τ_1^5 to τ_2^5; where $\mathbf{C}_5 \in \tau_0^5$, $\mathbf{C}_2 \in \tau_1^5$, $\mathbf{C}_1 \in \tau_2^5$, Δ_5 =FALSE, $\mathbf{RI}_5^2$=TRUE, Δ_2=FALSE and $\mathbf{RI}_2^1$=TRUE. The τ wave propagation from $\mathbf{C}_5$ to $\mathbf{C}_2$ implies that $\mathbf{C}_5$ is disassemblable after removing $\mathbf{C}_2$. Similarly, the τ wave propagation from $\mathbf{C}_2$ to $\mathbf{C}_1$ implies that $\mathbf{C}_2$ is disassemblable after removing $\mathbf{C}_1$.

A β wave determines the minimum number of components to be removed to disassemble $\mathbf{C}_i \in \mathbf{A}$. If $\mathbf{C}_i \in \beta_a$ then the minimum number of components to be removed to disassemble $\mathbf{C}_i$ is **a**. For example, Fig. 5 shows the β wave propagation for **A** in Fig. 3. $\mathbf{C}_1 \in \beta_1$, $\mathbf{C}_2 \in \beta_2$, Δ_1=TRUE, Δ_2=FALSE and $\mathbf{RI}_2^1$=TRUE. The minimal number of removals is 2 for $\mathbf{C}_2 \in \beta_2$ and 1 for $\mathbf{C}_1 \in \beta_1$.

2.4 Intersection Event. The intersection of τ and β waves (denoted as an Intersection Event, **IE**) determines **S** and it is defined as follows:

Definition 3 (*Intersection Event*): **IE** is an Intersection of any **m** ($1<=\mathbf{m}<=\mathbf{s}$) τ wave(s) ($\tau^{x1}, \tau^{x2}, \ldots, \tau^{xm}$; where $\mathbf{C}_{x1}, \mathbf{C}_{x2}, \ldots, \mathbf{C}_{xm} \in \mathbf{C}$) and a β wave at $\mathbf{C}_w \in \mathbf{A}$ (implies that $\mathbf{C}_w \in \tau^{x1}, \tau^{x2}, \ldots, \tau^{xm}, \beta$).

An **IE** is defined to determine both the disjoint (at $\mathbf{m}=1$) and common sequences (at $\mathbf{m}>1$) for target components. Let

$$\mathbf{C}_i \overset{P}{\sim} \mathbf{C}_j$$

denote a minimal-component removal sequence from $\mathbf{C}_i$ to $\mathbf{C}_j$. Every occurrence of an **IE** for $\mathbf{m}(>0)$ τ wave(s) ($\tau^{x1}, \tau^{x2}, \ldots, \tau^{xm}$) determines

$$\mathbf{S}=\{\mathbf{C}_b \overset{P}{\sim} \mathbf{C}_w, \mathbf{C}_w \overset{P}{\sim} \mathbf{C}_{x1}, \mathbf{C}_w \overset{P}{\sim} \mathbf{C}_{x2}, \ldots, \mathbf{C}_w \overset{P}{\sim} \mathbf{C}_{xm}\}$$

for $\mathbf{C}'=\{\mathbf{C}_{x1}, \mathbf{C}_{x2}, \ldots, \mathbf{C}_{xm}\} \subseteq \mathbf{C}$ and $\mathbf{C}_w \in \mathbf{A}$. The importance of the **IE** between waves lies in the determination of the component at which the waves intersect. Therefore the shape of the wave in the geometry space is irrelevant. The wave merely orders the components in topological space. For example, Fig. 6 shows the **RG** for $\mathbf{C}=\{\mathbf{C}_4, \mathbf{C}_5\}$ of **A** in Fig. 3. An **IE** occurs at $\mathbf{C}_2(\tau_1^5 \cap \tau_1^4 \cap \beta_2)$ with $\mathbf{m}=2$, which determines $\mathbf{S}=\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_4, \mathbf{C}_5\}$ with $\mathbf{n}_S=4$.

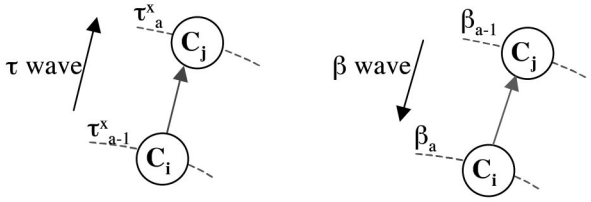

Fig. 4 τ wave and β wave propagation

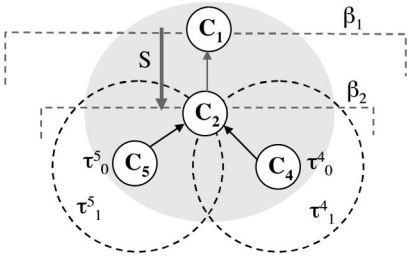

Fig. 6 RG for C={C4,C5}:A in Fig. 3

3 Wave Propagation Algorithms

Based on the **DWP** approach, two algorithms are proposed for **SD**. For $\mathbf{s} \ll \mathbf{n}$ target components *Multiple Wave Propagation* (**MWP**) Algorithm is presented. **MWP** defines time-based **IE**'s between disassembly waves and determines **OS** based on the order of event occurrence. However, for $\mathbf{s} < \mathbf{n}$ target components *Priority Intersection Event* (**PIE**) Algorithm is presented. **PIE** defines polynomial number of **IE**'s that are necessary candidate events in determining an **OS**. Both the algorithms determine locally optimum sequences in a feasible computation time.

3.1 Multiple Wave Propagation Algorithm. For **SD** of **s** components there are **s** τ waves and one β wave. Let **T** denote the time step. At $\mathbf{T}=0$, $\tau_0^{\mathbf{xk}}=\{\mathbf{C_{xk}}\}$ for $\mathbf{C_{xk}} \in \mathbf{C}(\mathbf{k}=1,\mathbf{s})$ and the β wave propagation is completed for all $\mathbf{C_i} \in \mathbf{A}$. For every time step (from $\mathbf{T}=\mathbf{a}$ to $\mathbf{T}=\mathbf{a}+1$; $\mathbf{a}>=0$), the $\tau^{\mathbf{xk}}$ propagates by one wave; i.e., at time step $\mathbf{T}=\mathbf{a}-1$, $\tau^{\mathbf{xk}}=(\tau_0^{\mathbf{xk}} \cup \tau_1^{\mathbf{xk}} \ldots \cup \tau_{\mathbf{a-1}}^{\mathbf{xk}})$ and at $\mathbf{T}=\mathbf{a}$, $\tau^{\mathbf{xk}}=(\tau_0^{\mathbf{xk}} \cup \tau_1^{\mathbf{xk}} \ldots \cup \tau_{\mathbf{a}}^{\mathbf{xk}})$. For every time step **T**, **IE**'s are determined between τ waves and β wave and the corresponding sequences are evaluated. At $\mathbf{T}=0$, disjoint sequences for every $\mathbf{C_{xk}} \in \mathbf{C}$ are determined. At $\mathbf{T}>0$, common sequences for $\mathbf{C}' \subseteq \mathbf{C}$ are determined. The evaluated sequences are then processed based on the order of event occurrence, i.e., by comparing every **S** with existing sequences for **C** based on minimal $\mathbf{n_S}$ to determine an **OS**.

To illustrate the **MWP** algorithm, consider **A** shown in Fig. 1 with $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$. The time-based **IE**'s and the wave propagation are illustrated in Fig. 7. The last column of the table shows the **S** for **C** at **T**. At $\mathbf{T}=3$, there are no more components for τ wave propagation, and therefore the **S** evaluated is an **OS**. For this example, the total number of **IE**'s is only 4, which can be computed in polynomial time.

However, the number of **IE**'s for **A** can be exponential, i.e., $O(2^s.\mathbf{n})$. This is due to the fact that all possible combinations of τ waves intersecting $\mathbf{C_i} \in \mathbf{A}$ must be checked for the occurrence of **IE**. For example, consider **A** in Fig. 8 with $\mathbf{C}=\{\mathbf{C_5},\mathbf{C_6},\mathbf{C_7}\}$. The RG at $\mathbf{T}=1$ is also shown in Fig. 8. At $\mathbf{T}=1$: τ^5, τ^6 and τ^7 waves from $\mathbf{C_5}$, $\mathbf{C_6}$, and $\mathbf{C_7}$ respectively, intersect β_1 at $\mathbf{C_4}$, i.e., $(\tau_1^5 \cap \tau_1^6 \cap \beta_1)$, $(\tau_1^5 \cap \tau_1^7 \cap \beta_1)$, $(\tau_1^6 \cap \tau_1^7 \cap \beta_1)$ and $(\tau_1^5 \cap \tau_1^6 \cap \tau_1^7 \cap \beta_1)$. Thus the number of checks necessary to determine all the **IE**'s at $\mathbf{C_4}$ is $O(\mathbf{2^s})$. Moreover, since $\mathbf{T}=O(\mathbf{n})$, the maximum number of **IE**'s is $O(\mathbf{2^s.n})$. For every **IE**, **S** is computed in $O(\mathbf{n})$ time. Therefore, the computational complexity of the **MWP** algorithm is $O(\mathbf{2^s.n^2})$. For a smaller number of target components $\mathbf{s} \ll \mathbf{n}$, the **MWP** algorithm is computationally efficient as compared to the enumeration approach.

3.2 Priority Intersection Event Algorithm. The **PIE** algorithm modifies the **MWP** algorithm in the definition and determination of **IE**. **PIE** prioritizes the **IE**'s and determines only the candidate events (denoted as ϕ events) for an **OS**.

Let at time step **T**, $\beta^{\mathbf{m}}=\beta_1 \cup \beta_2 \ldots \cup \beta_{(\mathbf{m*T})}$; $1<=\mathbf{m}<=\mathbf{s}$. For example, with $\mathbf{s}=2$, at $\mathbf{T}=0$: $\beta_0=\{\}$, $\beta^1=\beta^2=(\beta_0)$ and at $\mathbf{T}=1$: $\beta^1=(\beta_1)$, $\beta^2=(\beta_1 \cup \beta_2)$. The definition of ϕ events is as follows:

Definition 4 (ϕ events): Let

- Ψ = Set of all τ waves of $\mathbf{C}' \subseteq \mathbf{C}$ intersecting $\mathbf{C_w} \in \mathbf{A}$ at $\mathbf{T}>0$.
- μ = *Cardinality* (Ψ).
- $\Psi' \subseteq \Psi$. Every $\tau^{\mathbf{x}} \in \Psi'$ has not been intersected by a β^1 wave or $\tau_{\mathbf{T}}^{\mathbf{x}}$ intersects $\beta_{\mathbf{T}}$ at $\mathbf{T}>0$.
- ν = *Cardinality* (Ψ').

Then,

1. ϕ_1 *event:* intersection of a τ wave with β wave, where $\mathbf{T}=0$.
2. ϕ_2 *event:* intersection of all ν (>1) τ waves of Ψ' with a β^{ν} wave at $\mathbf{C_w}$, where $\mathbf{T}>0$.
3. ϕ_3 *event:* intersection of all μ (>1) τ waves of Ψ with a β^{μ} wave at $\mathbf{C_w}$ (where $\mathbf{T}>0$) such that the total number of waves from $(\mathbf{C}' \rightarrow \mathbf{C_w} \rightarrow \mathbf{C_b})$ is less than that of $(\mathbf{C}' \rightarrow \mathbf{C_b}$'s).

A ϕ_1 event for $\tau^{\mathbf{xk}}$ determines disjoint

$$\mathbf{OS}=\{\mathbf{C_b} \overset{P}{\sim} \mathbf{C_{xk}}\}$$

for $\mathbf{C_{xk}} \in \mathbf{C}$. To illustrate, consider **A** in Fig. 8 at $\mathbf{T}=0$: ϕ_1 events occur for τ^5, τ^6 and τ^7: $(\tau_0^5 \cap \beta_2)$ at $\mathbf{C_5}$, $(\tau_0^6 \cap \beta_2)$ at $\mathbf{C_6}$ and $(\tau_0^7 \cap \beta_2)$ at $\mathbf{C_7}$, respectively. Therefore, $\mathbf{OS}=\{\mathbf{C_4},\mathbf{C_5}\}$ for $\mathbf{C_5}$, $\mathbf{OS}=\{\mathbf{C_4},\mathbf{C_6}\}$ for $\mathbf{C_6}$ and $\mathbf{OS}=\{\mathbf{C_4},\mathbf{C_7}\}$ for $\mathbf{C_7}$, each with $\mathbf{n_S}=2$. Moreover, the ϕ_1 event for $\tau^{\mathbf{xk}}$ is better than other **IE**'s that determine disjoint **OS** for $\mathbf{C_{xk}} \in \mathbf{C}$.

A ϕ_2 event for ν (>1) τ waves: $\tau^{\mathbf{x1}}, \tau^{\mathbf{x2}}, \ldots, \tau^{\mathbf{x}\nu} \in \Psi'$ determines

$$\mathbf{S}=\{\mathbf{C_b} \overset{P}{\sim} \mathbf{C_w}, \mathbf{C_w} \overset{P}{\sim} \mathbf{C_{x1}}, \mathbf{C_w} \overset{P}{\sim} \mathbf{C_{x2}}, \ldots, \mathbf{C_w} \overset{P}{\sim} \mathbf{C_{x\nu}}\}$$

for $\mathbf{C}'=\{\mathbf{C_{x1}},\mathbf{C_{x2}},\ldots,\mathbf{C_{x\nu}}\} \subseteq \mathbf{C}$. The $\mathbf{n_S}$ for **S** from a ϕ_2 event is less than that for the disjoint **OS** for $\mathbf{C_{x1}},\mathbf{C_{x2}},\ldots,\mathbf{C_{xm}}$. For example, from the **RG** shown in Fig. 9, $\mathbf{S}=\{\mathbf{C_1},\mathbf{C_4},\mathbf{C_3},\mathbf{C_5}\}$ with $\mathbf{n_S}=4$ available from a ϕ_2 event for $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$ ($\mathbf{C_w}=\mathbf{C_4}$) is better than $\mathbf{S}=\{\mathbf{C_2},\mathbf{C_3},\mathbf{C_7},\mathbf{C_6},\mathbf{C_5}\}$ with $\mathbf{n_S}=5$ available from the

T	$\tau^3_{\mathbf{T}}$	$\tau^5_{\mathbf{T}}$	IE	S for C
0	$\mathbf{C_3}$	$\mathbf{C_5}$	$(\tau^3_0 \cap \beta_2)$, $(\tau^5_0 \cap \beta_3)$	$\{\mathbf{C_2, C_3, C_7, C_6, C_5}\}$
1	$\mathbf{C_2, C_4}$	$\mathbf{C_4, C_6}$	$(\tau^3_1 \cap \tau^5_1 \cap \beta_2)$	$\{\mathbf{C_1, C_4, C_3, C_5}\}$
2	$\mathbf{C_1}$	$\mathbf{C_1, C_7}$	$(\tau^3_2 \cap \tau^5_2 \cap \beta_1)$	$\{\mathbf{C_1, C_4, C_3, C_5}\}$
3	-	-	-	$\mathbf{OS} = \{\mathbf{C_1, C_4, C_3, C_5}\}$

Fig. 7 Illustration of the MWP algorithm for A in Fig. 1 with C $=\{\mathbf{C_3},\mathbf{C_5}\}$

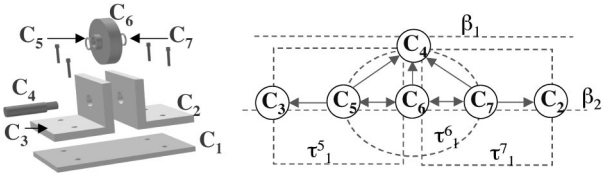

Fig. 8 Wheel support assembly (exploded view) and RG at T $=1$ for $\mathbf{C}=\{\mathbf{C_5},\mathbf{C_6},\mathbf{C_7}\}$

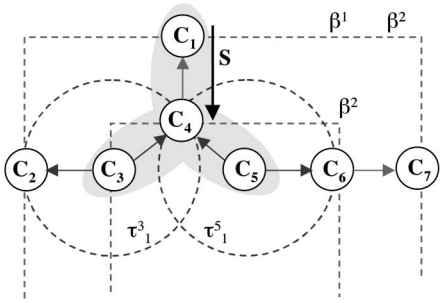

Fig. 9 RG at T=1 for $\mathbf{C}=\{\mathbf{C_3},\mathbf{C_5}\}$, A in Fig. 1: ϕ_2 event $\tau^3 \cap \tau^5 \cap \beta^2$ at $\mathbf{C_4}$

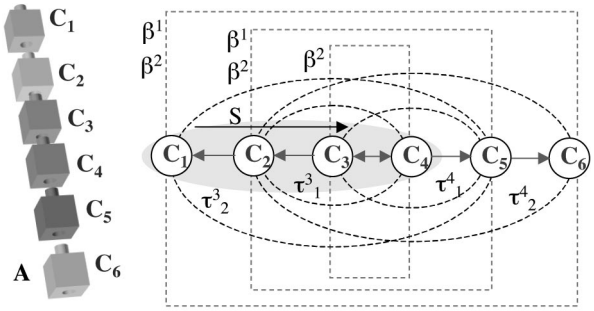

Fig. 10 Toy assembly (exploded view) and RG at T=2 for C $=\{C_3, C_4\}$

T	C'	C_w	ϕ events
0	C_5	C_5	ϕ_1: $(\tau^5{}_0 \cap \beta_2)$
	C_6	C_6	ϕ_1: $(\tau^6{}_0 \cap \beta_2)$
	C_7	C_7	ϕ_1: $(\tau^7{}_0 \cap \beta_2)$
1	C_5, C_6, C_7	C_4	ϕ_2: $(\tau^5 \cap \tau^6 \cap \tau^7 \cap \beta^3)$
	C_5, C_6	C_5	ϕ_2: $(\tau^5 \cap \tau^6 \cap \beta^2)$
	C_5, C_6, C_7	C_6	ϕ_2: $(\tau^5 \cap \tau^6 \cap \tau^7 \cap \beta^3)$
	C_6, C_7	C_7	ϕ_2: $(\tau^6 \cap \tau^7 \cap \beta^2)$

Fig. 11 Illustration of the PIE algorithm for A in Fig. 8 with C $=\{C_5, C_6, C_7\}$: T=0,1.

ϕ_1 events for $\{C_3, C_5\}$. Moreover, the ϕ_2 event that occurs for $C' \subseteq C$ at C_w is better than other **IE**s at C_w that occur for $C'' \subset C'$ at the same time step **T**.

A ϕ_3 event determines the sequence

$$\{C_b \overset{P}{\sim} C_w, C_w \overset{P}{\sim} C_{x1}, C_w \overset{P}{\sim} C_{x2}, \ldots, C_w \overset{P}{\sim} C_{x\mu}\}$$

for $C' = \{C_{x1}, C_{x2}, \ldots, C_{x\mu}\} \subseteq C$ and is a candidate for an **OS**. To illustrate this, consider the **RG** shown in Fig. 10 with $C = \{C_3, C_4\}$. At T=2, ϕ_3 event occurs at C_2: the intersection of τ^3, τ^4 and β^2. Clearly, $S = \{C_1, C_2, C_3, C_4\}$ with $n_S = 4$ available from this ϕ_3 event is better than $S = \{C_1, C_2, C_3, C_6, C_5, C_4\}$ with $n_S = 6$ available from ϕ_1 events.

Based on the above argument, ϕ events are necessary candidate events for an **OS** and include locally best events (events that are found to be optimal at time **T** for every C_w). For example, the ϕ events for **A** in Fig. 8 with $C = \{C_5, C_6, C_7\}$ are shown in Fig. 11. The total number of ϕ events for **A** is O(**sn**); i.e., polynomial. Therefore, the **PIE** algorithm is $O(s.n^2)$.

4 Results and Discussions

This section presents some of the results of the **DWP** approach, and discusses the contributions of the current research and future work.

For a Gear Reducer Assembly (Fig. 12), Fig. 13 shows the **RG** at T=2 for $C = \{C_2, C_3, C_4, C_{12}, C_{13}, C_{16}, C_{20}, C_{22}\}$. The **PIE** algorithm determines an $OS = \{C_1, C_2, C_3, C_4, C_{14}, C_{13}, C_{12}, C_{15}, C_{16}, C_{23}, C_{22}, C_{21}, C_{20}\}$ with $n_S = 13$, identified by $(\tau^2 \cap \tau^3 \cap \tau^4 \cap \beta^3)$, $(\tau^{12} \cap \tau^{13} \cap \beta^2)$, $(\tau_0^{16} \cap \beta_2)$ and $(\tau^{20} \cap \tau^{22} \cap \beta^2)$.

The **DWP** approach incorporates fasteners in **SD** analysis as follows: Determining **S** for **C** by ignoring the existence of fasteners $\{J_k\}$ and subsequently determining $\{J_k\}$ that need to be removed to disassemble all $C_i \in S$, and then modifying **S** appropriately [16]. For example, in Fig. 12 (with fasteners), $J_1, J_2, J_3, J_4 \in A$ should be removed to disassemble $C_1, C_{14}, C_{15}, C_{23} \in OS$, respectively. Therefore, the resultant $OS = \{J_1, C_1, C_2, C_3, C_4, J_2, C_{14}, C_{13}, C_{12}, J_3, C_{15}, C_{16}, J_4, C_{23}, C_{22}, C_{21}, C_{20}\}$ for $C = \{C_2, C_3, C_4, C_{12}, C_{13}, C_{16}, C_{20}, C_{22}\}$.

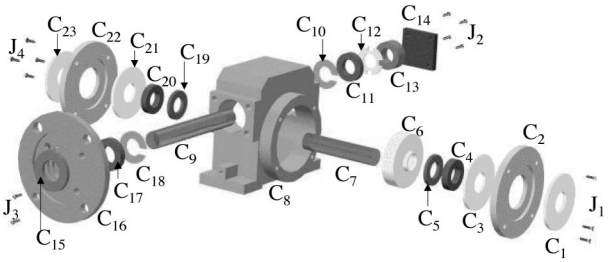

Fig. 12 Gear reducer assembly (exploded view)

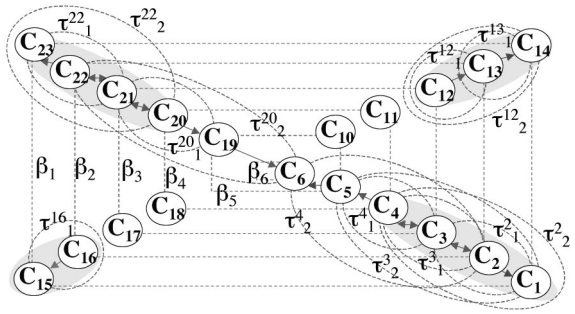

Fig. 13 RG at T=2 for A in Fig. 12, C $=\{C_2, C_3, C_4, C_{12}, C_{13}, C_{16}, C_{20}, C_{22}\}$.

Both **MWP** and **PIE** algorithms order events based on **T**; however, the events identified at each **T** and the number of such events differ. The **MWP** algorithm starts at T=0 with disjoint sequences for **C** and at T>0 tries to identify **S** that is better than previously computed **S**. Therefore, the **MWP** algorithm can be processed until some user-defined limiting time-step T′, determining a locally optimum **S** at T=T′. However, the **PIE** algorithm defines ϕ events with respect to C_w and **T**, thereby determining an **OS** only after all the ϕ events are processed. To illustrate, consider **A** in Fig. 10 with $C = \{C_3, C_4\}$: At T=1, $S = \{C_1, C_2, C_3, C_4\}$ with $n_S = 4$ by the **MWP** algorithm and $S = \{C_6, C_5, C_4, C_1, C_2, C_3\}$ with $n_S = 6$ by the **PIE** algorithm. However, the **PIE** algorithm determines $S = \{C_1, C_2, C_3, C_4\}$ at T=2. A comparison of **MWP** and **PIE** algorithms, based on the number of **IE**'s, is shown in Fig. 14.

A	n	C	s	# of events	
				MWP	PIE
Figure 1	9	{C3, C5}	2	4	4
Figure 3	7	{C4, C5}	2	4	4
Figure 8	7	{C5, C6, C7}	3	13	7
Figure 10	6	{C3, C4}	2	8	6
Figure 12	23	{C2, C3, C4, C12, C13, C16, C20, C22}	8	293	49

Fig. 14 MWP and PIE algorithms: performance comparison

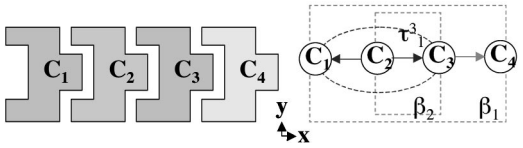

Fig. 15 Test assembly and RG for $C=\{C_2\}$

4.1 Contributions. The main contribution of this paper is a new approach, *Disassembly Wave Propagation*, for efficient multiple-component selective disassembly, and the two algorithms:

- *Multiple Wave Propagation* (**MWP**) algorithm for **SD** of $\mathbf{s} \leqslant \mathbf{n}$ components with computational complexity $O(\mathbf{n}^2\mathbf{2}^{\mathbf{s}})$.
- *Priority Intersection Event* (**PIE**) algorithm for **SD** of $\mathbf{s} < \mathbf{n}$ components with computational complexity $O(\mathbf{sn}^2)$.

4.2 Limitations and Future Work. One of the limitations of the current approach is that only the contact based disassembly is evaluated, which results in a local disassembly sequence. Therefore, the global interference is not checked. However, the **DWP** approach for multiple components may be extended for non-contact geometry as follows. Let the disassembly directions be predefined, e.g. $(\pm x, \pm y)$. The disassemblability of $\mathbf{C_i}$ is computed by collision detection. For example, in Fig. 15, Δ_2 = FALSE since $\mathbf{C_2}$ collides with $\mathbf{C_1}$ and $\mathbf{C_3}$. $\mathbf{RI}_2^1$ = TRUE since $\mathbf{C_2}$ does not collide with any other component along $-x$ direction. By utilizing the above concept, the geometric attributes are computed for **DWP** analysis. For example, Fig. 15 also shows the **RG** for **A**.

A second limitation is that the **DWP** approach for multiple **SD** is applied for the class of single dependent disassembly of components. Moreover, while the current research performs **SD** analysis with an objective of minimal component removals, other objectives such as minimal cost/time [5] have to be researched. One potential approach is first to determine a set of sequences, $\{\mathbf{S}\}$, from disassembly waves and then to determine OS that satisfies the required objective from $\{\mathbf{S}\}$ [3]. This approach allows an initial pruning of the solution space based on locally minimum removals and subsequent evaluation of an **OS** based on the required objective.

5 Summary

This paper proposes a new approach, disassembly wave propagation, for efficient selective disassembly of multiple components from a geometric model of an assembly. A minimal removal sequence to disassemble the target components is evaluated by determining and analyzing both the disjoint and common sequences between the target components. Two new algorithms are presented that determine candidate sequences for optimal sequencing in a feasible computation time.

Acknowledgments

This work was funded in part by the following grants: NSF research equipment grant (DMI 9622665), CenCITT grant (MTU 960238Z1) and Lucent Industrial Ecology Fellowship (26–96).

References

[1] Homem dé Mello, L. S., and Lee, S., 1991, *Computer Aided Mechanical Assembly Planning*, Kluwer Academic.
[2] Srinivasan, H., Shyamsundar, N., and Gadh, R., 1997, ''A Framework for Virtual Disassembly Analysis,'' J. Intell. Manufact., **8**, No. 4, pp. 277–295.
[3] Srinivasan, H., Figueroa, R., and Gadh, R., 1999, ''Selective Disassembly for Virtual Prototyping as Applied to De-Manufacturing,'' J. Robot. Comput. Integr. Manufact., **15**, No. 3, pp. 231–245.
[4] Anderson, D. M., 1990, *Design for Manufacturability—Optimizing Cost, Quality and Time-to-Market*, CIM Press, California.
[5] Boothroyd, G., Dewhurst, P., and Knight, W., 1994, *Product Design for Manufacturing and Assembly*, Marcel Dekker Inc.
[6] Wolter, J. D., 1999, ''*Assembly Sequence Planning Bibliography*,'' http://www.cs.tamu.edu/research/robotics/Wolter/asp/bib.html.
[7] Baldwin, D. F., Abel, T. E., Lui, M. C. M., De-Fazio, T. L., and Whitney, D. E., 1991, ''An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products,'' IEEE Trans. Rob. Autom., **7**, No. 1, pp. 78–94.
[8] Lee, S., and Shin, Y. G., 1990, ''Assembly Planning Based on Geometric Reasoning,'' Comput. Graph., **14**, No. 2, pp. 237–250.
[9] Wilson, R. H., Kavraki, L., Lozano-Perez, T., and Latombe, J. C., 1995, ''Two Handed Assembly Sequencing,'' Int. J. Robot. Res., **14**, No. 4, pp. 335–350.
[10] Beasley, D., and Martin, R. R., 1993, ''Disassembly Sequences for Objects Built from Unit Cubes,'' CAD J., **25**, No. 12, pp. 751–761.
[11] Mattikalli, R. S., and Khosla, P., 1989, ''Determining the Assembly Sequence from a 3-D Model,'' J. Mech. Work. Technol., **20**, pp. 153–162.
[12] Woo, T. C., and Dutta, D., 1991, ''Automatic Disassembly and Total Ordering in Three Dimensions,'' ASME J. End. Ind., **113**, No. 1, pp. 207–213.
[13] Dutta, D., and Woo, A. C., 1992, ''Algorithms for Multiple Disassembly and Parallel Assemblies,'' ASME Conc. Eng., **PED-59**, pp. 257–266.
[14] Goldwasser, M., Latombe, J. C., and Motwani, R., 1996, ''Complexity Measures for Assembly Sequences,'' *Proc. of the IEEE Int. Conference on Robotics and Automation*, pp. 1581–1587.
[15] Kaufmann, S. G., Wilson, R. H., Jones, R. E., Calton, T. L., and Ames, A. L., 1996, ''The Archimedes Mechanical Assembly Planning System,'' *Proc. of the IEEE Int. Conference on Robotics and Automation*, pp. 3361–3368.
[16] Srinivasan, H., and Gadh, R., 1998, ''A Geometric Algorithm for Single Selective Disassembly Using the Wave Propagation Abstraction,'' CAD J., **30**, No. 8, pp. 603–613.